Package 'MaximinInfer'

November 8, 2021

Type Package

Title Inference for Maximin Effect in High-Dimensional Settings

Version 1.0.0

Author Zhenyu Wang; Zijian Guo

Maintainer Zhenyu Wang <wangzy0701@gmail.com>

Description Implementation of the sampling and aggregation method for the covariate shift maximin effect, which was proposed in <arXiv:2011.07568>. It constructs the confidence interval for any linear combination of the high-dimensional maximin effect.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

Imports MASS, stats, CVXR, glmnet, intervals, SIHR

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2021-11-08 12:10:15 UTC

R topics documented:

`decide_delta` *decide delta data-dependently*

Description

`decide_delta` will tell if the estimator is stable or not without ridge penalty at first. If instable, it picks a ridge penalty data-dependently.

Usage

```
decide_delta(
  object,
  gen.size = 500,
  step_delta = 0.1,
  MAX_iter = 100,
  verbose = FALSE
)
```

Arguments

`object`	Object of class inheriting from "Maximin"
`gen.size`	The generating sample size (Default = 500)
`step_delta`	The step size of searching delta (Default = 0.1)
`MAX_iter`	Maximum of iterations for searching (Default = 100)
`verbose`	Print information about delta and reward (Default = 'FALSE')

Value

`delta`	The data-dependent ridge penalty
`reward.ratio`	The ratio of penalized reward over non-penalized reward

Examples

```
## The problem is low-dimensional for testings
## heterogenous data and covariates shift
X1 = sample_data$X1
X2 = sample_data$X2
Y1 = sample_data$Y1
Y2 = sample_data$Y2
X.target = sample_data$X.target

## loading
loading = rep(0, 5) # dimension p=5
loading[5] = 1

## call
mm <- Maximin(list(X1, X2), list(Y1, Y2), loading, X.target)
```

```
out <- decide_delta(mm, gen.size=5)
out$delta
out$reward.ratio
```

`infer` *Inference method for class "Maximin"*

Description

Point estimator and Confidence interval based on Maximin object

Usage

```
infer(
  object,
  delta = 0,
  gen.size = 500,
  threshold = 0,
  alpha = 0.05,
  alpha.thres = 0.01
)
```

Arguments

`object`	Object of class inheriting from "Maximin"
`delta`	The ridge penalty (Default = 0)
`gen.size`	The generating sample size (Default = 500)
`threshold`	Should generated samples be filtered or not? if 0, use normal threshold to filter; if 1, use chi-square threshold to filter; if 2, do not filter (Default = 0)
`alpha`	confidence value to construct confidence interval (Default = 0.05)
`alpha.thres`	confidence value to select generated samples (Default = 0.01)

Value

`weight`	The weight vector for groups, of length L
`point`	The point estimator of the linear combination
`mm.effect`	The aggregated maximin effect (coefficients), of length p or $p+1$
`CI`	Confidence interval for the linear combination

Examples

```
## The problem is low-dimensional and we do sampling only 5 times instead of 500 for testings
## heterogenous data and covariates shift
X1 = sample_data$X1
X2 = sample_data$X2
Y1 = sample_data$Y1
Y2 = sample_data$Y2
X.target = sample_data$X.target

## loading
loading = rep(0, 5) # dimension p=5
loading[5] = 1

## call
mm <- Maximin(list(X1, X2), list(Y1, Y2), loading, X.target, covariate.shift = TRUE)
mmInfer <- infer(mm, gen.size=5)
```

`Maximin` *Class Maximin*

Description

'Maximin' returns the class "Maximin", which provides materials for later inference method.

Usage

```
Maximin(
  Xlist,
  Ylist,
  loading,
  X.target = NULL,
  cov.target = NULL,
  covariate.shift = TRUE,
  lam.value = c("CV", "CV.min"),
  intercept = TRUE,
  intercept.loading = FALSE
)
```

Arguments

`Xlist`	list of design matrix for source data, of length L
`Ylist`	list of outcome vector for source data, of length L
`loading`	Loading, of length p
`X.target`	Design matrix for target data, of dimension $n.target$ x p (default = 'NULL')
`cov.target`	Covariance matrix for target data, of dimension p x p (default = 'NULL')
`covariate.shift`	Covariate shifts or not between source and target data (default = 'TRUE')

lam.value	The method to be used to obtain Lasso estimator of high-dimensional regression vector for each group
intercept	Should intercept be fitted for the initial estimator (default = 'TRUE')
intercept.loading	Should intercept be included for the loading (default = 'FALSE')

Details

The algorithm implemented scenarios with or without covariate shift. If 'cov.target' is specified, the 'X.target' will be ignored; if not, while 'X.target' is specified, 'cov.target' will be estimated by 'X.target'. If both are not specified, the algorithm will automatically set 'covariate.shift' as 'FALSE'.

Value

'Maximin' returns an object of class "Maximin". The function 'infer' is used to do further inference. An object of class "Maximin" is a list containing the following components.

Gamma.prop	The proposed debiased regression covariance matrix
Coef.est	matrix, of dimension $p(+1)$ x L where each column corresponds to the Lasso estimator of the high-dimensional regression vector for a given group
Point.vec	vector, of length L with the l-th entry as the debiased estimator of the linear combination of the l-th high-dimensional regression vector
L	The number of groups
gen.mu	The mean vector for sampling the regression covariance matrix
gen.Cov	The variance matrix for sampling the regression covariance matrix

---

MaximinInfer *The Wrapper function for Maximin inference*

Description

MaximinInfer is a wrapper for class Maximin and the method infer.

Usage

```
MaximinInfer(
  Xlist,
  Ylist,
  loading,
  X.target = NULL,
  cov.target = NULL,
  covariate.shift = TRUE,
  lam.value = c("CV", "CV.min"),
  intercept = TRUE,
  intercept.loading = FALSE,
```

```
  delta = 0,
  gen.size = 500,
  threshold = 0,
  alpha = 0.05,
  alpha.thres = 0.01
)
```

Arguments

`Xlist`	list of design matrix for source data, of length L
`Ylist`	list of outcome vector for source data, of length L
`loading`	Loading, of length p
`X.target`	Design matrix for target data, of dimension $n.target$ x p (default = 'NULL')
`cov.target`	Covariance matrix for target data, of dimension p x p (default = 'NULL')
`covariate.shift`	Covariate shifts or not between source and target data (default = 'TRUE')
`lam.value`	The method to be used to obtain Lasso estimator of high-dimensional regression vector for each group
`intercept`	Should intercept be fitted for the initial estimator (default = 'TRUE')
`intercept.loading`	Should intercept be included for the loading (default = 'FALSE')
`delta`	The ridge penalty (Default = 0)
`gen.size`	The generating sample size (Default = 500)
`threshold`	Should generated samples be filtered or not? If 0, use normal threshold to filter; if 1, use chi-square threshold to filter; if 2, do not filter. (Default = 0)
`alpha`	confidence value to construct confidence interval (Default = 0.05)
`alpha.thres`	confidence value to select generated samples (Default = 0.01)

Details

The algorithm implemented scenarios with or without covariate shift. If 'cov.target' is specified, the 'X.target' will be ignored; if not, while 'X.target' is specified, 'cov.target' will be estimated by 'X.target'. If both are not specified, the algorithm will automatically set 'covariate.shift' as 'FALSE'.

Value

`weight`	The weight vector for groups, of length L
`point`	The point estimator of the linear combination
`mm.effect`	The aggregated maximin effect (coefficients), of length p or $p+1$
`CI`	Confidence interval for the linear combination

Examples

```
## The problem is low-dimensional and we do sampling only 5 times instead of 500 for testings
## heterogenous data and covariates shift
X1 = sample_data$X1
X2 = sample_data$X2
Y1 = sample_data$Y1
Y2 = sample_data$Y2
X.target = sample_data$X.target

## loading
loading = rep(0, 5) # dimension p=5
loading[5] = 1

## call
mmInfer <- MaximinInfer(list(X1, X2), list(Y1, Y2), loading, X.target, gen.size=5)
```

---

measure_instability *measurement of instability*

Description

compute the instability measurement given a specific ridge penalty

Usage

```
measure_instability(
  object,
  delta = 0,
  gen.size = 500,
  threshold = 0,
  alpha.thres = 0.01
)
```

Arguments

object	Object of class inheriting from "Maximin"
delta	The ridge penalty (Default = 0)
gen.size	The generating sample size (Default = 500)
threshold	Should generated samples be filtered or not? if 0, use normal threshold to filter; if 1, use chi-square threshold to filter; if 2, do not filter. (Default = 0)
alpha.thres	confidence value to select generated samples (Default = 0.01)

Value

measure	The measurement of instability

Examples

```
## The problem is low-dimensional and we do sampling only 5 times instead of 500 for testings
## heterogenous data and covariates shift
X1 = sample_data$X1
X2 = sample_data$X2
Y1 = sample_data$Y1
Y2 = sample_data$Y2
X.target = sample_data$X.target

## loading
loading = rep(0, 5)
loading[5] = 1

## call
mm <- Maximin(list(X1, X2), list(Y1, Y2), loading, X.target, covariate.shift = TRUE)
out <- measure_instability(mm, gen.size=5)
out$measure
```

`sample_data` *Sample Data for Analysis*

Description

Datasets for the simple testing and running examples. The data is heterogenous with 2 groups, and covariates shift between target data and source data.

Usage

```
sample_data
```

Format

list with source data and target data, which are:

X1 Design matrix for the 1st group source data

X2 Design matrix for the 2nd group source data

Y1 Outcome vector for the 1st group source data

Y2 Outcome vector for the 2nd group source data

X.target Design matrix for the target data

Index